REMARKS

Applicant thanks the examiner for his time and comments during the telephone interview held on November 20, 2006, and attended by Examiner Huisman and applicant's undersigned attorney, Ido Rabinovitch. During the Examiner's Interview, the claims and the cited art were discussed. No agreement regarding the claims was immediately reached.

Claims 1-38 are pending. Claims 1 and 20 are independent.

Applicant acknowledges with thanks the examiner's indication that claims 7, 12, 15, 18, 19, 26, 31, 34, 37 and 38 would be allowable.

The examiner rejected claims 21-23 under 35 U.S.C. §112, second paragraph, on the ground that there is no antecedent basis for the term "the register field" appearing in claim 21.

In response, and in accordance with the examiner's request, applicant amended claim 21 to replace the wording "the register field" with "the register."

The examiner rejected claims 1-2, 5-6, 8-10, 20-21, 24-25 and 27-29 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,697,935 to Borkenhagen et al. The examiner further rejected claims 3-4, 11, 13-14, 16-17, 22-23, 30, 32-33 and 35-36 under 35 U.S.C. 103(a) as being unpatentable over Borkenhagen.

Specifically, with respect to independent claim 1, the examiner stated:

> 8. Referring to claim 1, Borkenhagen has taught a method of operating a processor comprising:
> a) receiving data specified by execution of a fast-write instruction in a processing thread identified by a processing thread number. See column 14, lines 3-18. Note that an instruction is used to set or clear bits in the thread switch control register. Data is received which will set or clear the bits in the register. Furthermore, since Borkenhagen talks about thread switching, there are multiple threads in the system. See column 4, lines 29-32, and note the existence of a thread number, which must inherently exist. That is, the system must know which thread is the 1st thread, the 2nd thread, the 3rd thread, etc. Otherwise, there would be no way for the system to switch among threads.
> b) selecting bit positions of a register specified by execution of the fast-write instruction according to the processing thread number. See column 14, lines 3-13. Note that in one embodiment, multiple thread switch control registers (one for each thread) may be implemented for more flexibility (i.e., different threads may be switched out for different reasons). With multiple registers, when a thread is to execute an

> instruction to modify its control register, the appropriate control register is specified according to the processing thread number. For instance, if the first thread is executing, then the first control register is selected. It should be noted that applicant has not claimed selecting some portion of bits in a single register where the selecting is based on the thread number,
> c) loading the data into the selected bit positions of the register. See column 14, lines 3-64. (Office Action, pages 3-4)

Applicant amended independent claim 1 to clarify that the register specified by the fast-write instruction includes multiple groups of bits, each group of bits associated with one of multiple threads available on the processor, and to further clarify that the processing thread number is used to select a group of bits associated with the processing thread. Support for this clarification is found, for example, at page 10, lines 17-24 of the originally filed application. Applicant similarly amended independent claim 20.

Applicant's amended independent claim 1 thus recites "receiving data specified by execution of a fast-write instruction in a processing thread identified by a processing thread number, the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with one of multiple threads available on the processor; selecting a group of bits associated with the processing thread, the group of bits being selected from the multiple groups of bits of the register specified by the fast-write instruction according to the processing thread number; and loading the data into the bit positions of the selected group of bits of the register." Accordingly, which particular group of bits, from the multiple groups of bits of a register (e.g., a control and status register), is loaded with the received data depends on the particular processing thread number.

In contrast, Borkenhagen describes a method and apparatus for selecting thread switch events in a multithreaded processor. Particularly, Borkenhagen's apparatus includes thread switch logic hardware 400 that determines whether a thread will be switched and, if so, what thread (FIG. 4, col. 10, lines 5-7). Borkenhagen explains that thread switching can be based on occurrence of certain events. For example, a cache miss can cause thread switching operation to take place (col. 10, lines 15-32). To that end, Borkenhagen describes that "[t]hread switch logic 400 comprises a thread switch control register 410 which controls what events will result in a

thread switch. For instance, the thread switch control register 410 can block events that cause state changes from being seen by the thread switch controller 450 so that a thread may not be switched as a result of a blocked event" (col. 10, lines 39-45). Borkenhagen explains, in relation to the thread switch control register 410, that:

> **The thread switch control register 410 is a software programmable register which selects the events to generate thread switching and has a separate enable bit for each defined thread switch control event. Although the embodiment described herein does not implement a separate thread switch control register 410 for each thread, separate thread switch control registers 410 for each thread could be implemented to provide more flexibility and performance at the cost of more hardware and complexity. Moreover, the thread switch control events in one thread switch control register need not be identical to the thread switch control events in any other thread switch control register.**
>
> **The thread switch control register 410 can be written by a service processor with software such as a dynamic scan communications interface disclosed in U.S. Pat. No. 5,079,725 entitled Chip Identification Method for Use with Scan Design Systems and Scan Testing Techniques or by the processor itself with software system code. The contents of the thread switch control register 410 is used by the thread switch controller 450 to enable or disable the generation of a thread switch. A value of one in the register 410 enables the thread switch control event associated with that bit to generate a thread switch. A value of zero in the thread switch control register 410 disables the thread switch control event associated with that bit from generating a thread switch. Of course, an instruction in the executing thread could disable any or all of the thread switch conditions for that particular or for other threads. The following table shows the association between thread switch events and their enable bits in the register 410. (col. 14, lines 3-30)**

Thus, Borkenhagen's thread switch control register 410 specifies the type of events that can be used to control general thread switching. Although, as the examiner observed, different registers may be used with respect to different threads to, presumably, individually control which events will affect thread switching for individual threads, at no point does Borkenhagen describe that a particular register includes groups of bits that are associated with different available threads. Borkenhagen certainly does not describe that a thread number, such as the currently processing thread number, is used to select a specific group of bits from multiple groups of bits of the register. Accordingly, Borkenhagen does not disclose or suggest none of the features of "receiving data specified by execution of a fast-write instruction in a processing thread identified by a processing thread number, the fast-write instruction further specifying a register, the register

having multiple groups of bits, each group of bits associated with one of multiple threads available on the processor," "selecting a group of bits associated with the processing thread, the group of bits being selected from the multiple groups of bits of the register specified by the fast-write instruction according to the processing thread number" or "loading the data into the bit positions of the selected group of bits of the register," as required by applicant's independent claim 1. Independent claim 1 is therefore patentable over the cited art.

Claims 2-19 depend from independent claim 1, and are therefore patentable over the cited art for at least the same reasons as independent claim 1.

Independent claim 20 recites the features of "receive data specified in the fast-write instruction in a processing thread identified by a processing thread number, the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with one of multiple threads available on the computer; select a group of bits associated with the processing thread, the group of bits being selected from the multiple groups of bits of the register specified in the fast-write instruction according to the processing thread number; and load the data into the bit positions of the selected group of bits of the register." For reasons similar to those provided with respect to independent claim 1, at least these features are not disclosed by the cited art. Accordingly, independent claim 20 is patentable over the cited art. Claims 21-38 depend from independent claim 20 and are therefore patentable for at least the same reasons as independent claim 20.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

In view of the foregoing remarks, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made

arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

Please apply any required fees to deposit account 06-1050, referencing the attorney docket number shown above.

Respectfully submitted,

Date: November 21, 2006

Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21486933.doc